Temporal-wise Attention Spiking Neural Networks for Event Streams Classification

Man Yao*, Huanhuan Gao*, Guangshe Zhao†, Dingheng Wang1, Yihan Lin2, Zhaoxu Yang1, Guoqi Li†
1Xi’an Jiaotong University 2Tsinghua University
{zhaogs@mail.xjtu.edu.cn, liguoqi@tsinghua.edu.cn}

Abstract

How to effectively and efficiently deal with spatio-temporal event streams, where the events are generally sparse and non-uniform and have the μs temporal resolution, is of great value and has various real-life applications. Spiking neural network (SNN), as one of the brain-inspired event-triggered computing models, has the potential to extract effective spatio-temporal features from the event streams. However, when aggregating individual events into frames with a new higher temporal resolution, existing SNN models do not attach importance to the different signal-to-noise ratios since event streams are sparse and non-uniform. This situation interferes with the performance of existing SNNs. In this work, we propose a temporal-wise attention SNN (TA-SNN) model to learn frame-based representation for processing event streams. Concretely, we extend the attention concept to temporal-wise input to judge the significance of frames for the final decision at the training stage, and discard the irrelevant frames at the inference stage. We demonstrate that TA-SNN models improve the accuracy of event streams classification tasks. We also study the impact of multiple-scale temporal resolutions for frame-based representation. Our approach is tested on three different classification tasks: gesture recognition, image classification, and spoken digit recognition. We report the state-of-the-art results on these tasks, and get the essential improvement of accuracy (almost 19%) for gesture recognition with only 60 ms.

1. Introduction

Dynamic vision sensors (DVS)[20, 28] pose a new paradigm shift by using sparse and asynchronous events to represent visual information. Unlike the conventional cameras, which produce fixed low-rate synchronized frames (typically less than 60 frames per second), DVS cameras encode the time, location, and polarity of the brightness changes for each pixel at an extremely high event rate (1M to 1G events per second), and exhibit advantages mainly in three aspects[11, 30]. Firstly, DVS cameras require much less resource, as the events are sparse and only triggered when the intensity changes. Secondly, the μs temporal resolution (TR) of DVS can avoid motion blur by producing high-rate events. Thirdly, DVS cameras have a high dynamic range (140dB vs. 60dB of conventional cameras), which makes them able to acquire information from challenging illumination conditions. These characteristics bring superiorities over conventional cameras when orienting to visual tasks which need low latency, low power consumption, and stability for variant illumination, which have been used in high-speed object tracking[30], autonomous driving[4], SLAM[7], low-latency interaction[1], etc.

However, we have observed that the event streams

* Equal contribution.
† Corresponding authors.
recorded by DVS cameras are usually redundant in the temporal dimension, which is caused by high TR and irregular dynamic scene changes. This characteristic makes event streams almost impossible to process directly through deep neural networks (DNNs), which are based on dense computation. Compromising on this, additional data preprocessing[2, 33, 11] is required and inevitably dilutes the advantages of low-latency and power-saving of events. Inspired by the working pattern of the mammalian visual cortex[11], spiking neural networks (SNNs) have a unique event-triggered computation characteristic that can respond to the events in a nearly latency-free and power-saving way[31, 25, 11], and it is naturally fit for processing event. However, due to the lack of training technology, the performance of deep SNNs has become the biggest obstacle to their application. During experiments, we find that there is still a lot of optimizing room for SNNs to process the event data more efficiently and effectively. That’s why we introduce the attention mechanism into SNNs.

In this work, we propose the temporal-wise attention SNNs (TA-SNNs) by extending the attention concept to temporal-wise input to automatically filter out the irrelevant frames for the final decision. For TA-SNNs, how to implement the attention mechanism while retaining the event-triggered characteristic is the primary consideration. Classic attention methods, such as self-attention[36], are hard to use because the change of network connection destroys the event-triggered characteristic in SNNs. Inspired by squeeze-and-excitation (SE) block[13], we design the TA module to obtain the statistical features of events at different times, generate the attention scores and then weigh the events by the scores. At the same time, we propose a data augmentation method called random consecutive slice (RCS) to utilize the event data. In order to keep the event-encoded data characteristics, we then use binary attention scores at the inference stage with a threshold in the TA module, which is termed as input attention pruning (IAP) and obtains an unchanged or even higher accuracy with RCS. Without losing generality, we test our approach with two kinds of SNN models, i.e., leaky integrate-and-fire (LIF) and leaky integrate-and-analog-fire (LIAF), on three kinds of tasks: gesture recognition, image classification, and spoken digit recognition. We report the state-of-the-art results on these tasks in long-term TRs, and get the essential improvement of accuracy (almost 19%) for gesture recognition with low-latency and power-saving property.

We summarize our contributions as follows:

1) We propose the TA-SNNs for event streams that can undertake the end-to-end training and inference tasks with low latency, low power consumption, and high performance. To the best of our knowledge, this is the first work to introduce temporal-wise attention into SNNs.

2) We propose the IAP method for SNNs and get similar or even better performance compared with those using full inputs (see Fig.4). The IAP brings a crucial power-saving significance for SNNs and other event-based algorithms.

3) Inspired by the data augmentation method in video recognition[35] and overlap method for event stream process[21], we introduce the RCS method to make full use of the sampled data.

2. Related Works

Event Streams Classification. To yield sufficient signal-to-noise ratios (SNR) for the task accuracy, processing the events as groups is the most common method[11]. In this paper, we adopt the frame-based representation that aggregates event streams into frames[27]. The frame-based representation is easy to generate and naturally compatible with the traditional computer vision framework, and the SNN algorithms based on frames can be easily mapped to neuromorphic hardware[27]. TR is a crucial parameter for frame-based representation, and generally, the bigger TR is, the higher SNR we could have. Most related works are dedicated to using various techniques to improve the classification performance based on the long-term TR, such as improve training method[45, 39], change the connection path of the SNNs[5, 42], and hybrid fusion[40, 8, 17], etc.

Spiking Neural Networks. Spiking neurons, such as LIF[38], use spike stream as the data transmission form and connect each other hierarchically as a network, i.e., SNNs. One common way in these spike-based SNNs is to assume that the neurons which have not received any input spikes will skip computations, i.e., event-triggered characteristic[11]. So spike-based SNNs can extract information from spikes in a power-saving way. The other kind of SNNs, i.e., analog-based SNNs, use dynamic characteristics in spiking neurons but transmit analog values in the network, such as LIAF[40], RELU SRNN[6], SpArNet[15], etc. Analog value makes the network easy to train, but it loses the attributes of skipping computation in spike-based SNNs. Without loss of generality, we separately adopt LIF and LIAF models as the elements of spike-based and analog-based SNNs to test the attention mechanism.

Attention Models. The attention mechanism selectively focuses on the most informative components of the input and can be interpreted as the sensitivity of the output to the variant input[34]. The models using attention have been applied to many tasks, such as sequence learning[26, 22], machine translation[36, 9, 37], action recognition[14, 10], etc. Generally, there are two types of works, i.e., temporal-wise attention in RNNs[34] and spatial-wise attention in SNNs[3, 41, 18] may be related to the proposed method in this paper. Our work is different from prior works, and we
focus on the statistical characteristics of the frames input at different timesteps based on SNNs.

3. Model Description

3.1. Frame-based Representation

Event steam comprises four dimensions: two spatial coordinates, the timestamp, and the polarity of the event. The polarity indicates an increase (ON) or decrease (OFF) of brightness, where ON/OFF can be represented via +1/-1 values. Assume the TR of event stream is \( \tau \). For frame, set a new TR \( \tau' = \tau \) as the input at timestep \( t \), and the consecutive \( \beta \) spike patterns can be grouped as a set

\[
E_t = \{ X_{t'} \}
\]

where \( t' \in [\beta \times t, \beta \times (t + 1) - 1] \) and \( \beta \) is called resolution factor. Then, the frame of input layer at time \( X_{t,n}^t \in R^{L \times B \times 2} \) based on \( dt \) can be got by

\[
X_{t,n}^0 = q(E_t)
\]

where \( t \in \{ 1, 2, \cdots, T \} \) is timestep, and aggregation function \( q(\cdot) \) could be selected, as non-polarity aggregation[21], accumulate aggregation[8], AND logic operation aggregation[12], etc. Here we choose a simple approach, which accumulates event stream with the information of event polarity.

3.2. Spiking Neural Network Models

The LIF model is a trade-off between the complex dynamic characteristics of biological neurons and the simpler mathematical form. It is suitable for simulating large-scale SNNs and can be described by a differential function[31]

\[
\tau \frac{du(t)}{dt} = -u(t) + I(t)
\]

where \( \tau \) is a time constant, and \( u(t) \) and \( I(t) \) are the membrane potential of the postsynaptic neuron and the input collected from presynaptic neurons, respectively (see the relationship of \( u(t) \) and \( I(t) \) in Fig.2). For easy inference and training, a simple iterative representation of LIF model[23] or LIAF model[40] can be described as

\[
\begin{align*}
U_{t,n}^t &= H_{t-1,n}^t + g(W^n, X_{t,n-1}^t) \\
Z_{t,n}^t &= f(U_{t,n}^t - u_{th}) \\
H_{t,n}^t &= \left( e^{-\frac{1}{\tau}} U_{t,n}^t \right) \circ \left( 1 - Z_{t,n}^t \right) \\
X_{t,n}^t &= \left\{ \begin{array}{ll} 
Z_{t,n}^t & \text{for LIF}, \\
\text{ReLU} (U_{t,n}^t) & \text{for LIAF}, 
\end{array} \right.
\end{align*}
\]

where \( n \) and \( t \) are indices of layer and timestep, \( W^n \) is the synaptic weight matrix between two adjacent layers, \( g(\cdot) \) is a function stands for convolutional operation or fully connected operation. \( f(\cdot) \) is a Heaviside step function that satisfies \( f(x) = 1 \) when \( x \geq 0 \), otherwise \( f(x) = 0 \). \( u_{th} \) is the membrane potential threshold, \( e^{-\frac{1}{\tau}} \) reflects the leakage factor of the membrane potential, \( \circ \) is the Hadamard product, \( X \) and \( H \) are spatial and temporal input, respectively, \( U \) is the membrane potential, and \( Z \) is the spike tensor. For spatial input tensor \( X \), its representation is different for LIF and LIAF which are separately described below.

LIF-SNNs. As shown in Eq.4 and Fig.2, by coupling \( X_{t,n-1}^t \) from the \( n-1 \) layer and \( H_{t-1,n}^t \) from the \( t-1 \) timestep, we can get \( U_{t,n}^t \). If \( U_{t,n}^t \) is greater than \( u_{th} \), the neuron executes the fire mechanism, which outputs \( Z_{t,n}^t \) as the spatial input of next layer, i.e., \( X_{t,n}^t = Z_{t,n}^t \), and resets \( U_{t,n}^t \) to \( u_{rest} \). Meanwhile, the neuron executes the leak mechanism, and the decayed value of membrane potential \( H_{t,n}^t \) will be used as the temporal input for the next timestep.

LIAF-SNNs. For the LIAF, it keeps the \( H_{t-1,n}^t \) and changes the Heaviside step function to ReLU function for \( U_{t,n}^t \), i.e., \( X_{t,n}^t = \text{ReLU} (U_{t,n}^t) \), then both spatial and temporal domains are analog values. We use the STBP[38] and the BPTT algorithm[40] to train LIF-SNNs and LIAF-SNNs, respectively.

3.3. Temporal-wise Attention for SNNs

The goal of TA module is to estimate the saliency of each frame. This saliency score should not only be based on the input statistical characteristic at the current timestep, but also take into consideration the information from neighboring frames. We apply the squeeze step and excitation step[13] in temporal-wise to implement the above two points. The spatial input tensor of \( n \)-th layer at \( t \)-th timestep is \( X_{t,n-1}^t \in R^{L \times B \times C} \) where \( C \) is channel size.

Squeeze step calculates a statistical vector of event numbers, and the value of statistical vector \( s_{n-1}^t \in R^T \) at \( t \)-th timestep is

\[
s_{n-1}^t = \frac{1}{L \times B \times C} \sum_{k=1}^{C} \sum_{i=1}^{L} \sum_{j=1}^{B} X_{k,i,j}^t \times s_{n-1}^{t-1}(k,i,j).
\]

By executing the excitation step, \( s_{n-1}^t \) is subjected to non-linear mapping through a two-layer fully connected network to obtain the correlation between different frames, i.e., score vector

\[
d_{n-1}^t = \left\{ \begin{array}{ll} 
\sigma (W_1^t \delta (W_2^t s_{n-1}^t)) & \text{training}, \\
\sigma (W_2^t \delta (W_1^t s_{n-1}^t)) - d_{th} & \text{inference}, 
\end{array} \right.
\]

where \( \delta \) and \( \sigma \) are ReLU and sigmoid activation function, respectively, \( W_1^t \in R^{2 \times T} \) and \( W_2^t \in R^{T \times 2} \) are trainable parameter matrices, and optional parameter \( r \) is used to control the model complexity, \( f(\cdot) \) is a Heaviside step func-
tion that is same as in Eq.4, and $d_{th}$ is the attention threshold. We use the score vector to train a complete network at the training stage. As an optional operation, at the inference stage, we discard the irrelevant frames which are lower than $d_{th}$, and set the attention score of the other frames to 1.

Finally, we use $d^{n-1}$ as the input score vector, and the final input at $t$th timestep is

$$\bar{X}^t_{t,n-1} = d^{t-1}_{t} X_{t,n-1}$$

where $\bar{X}^t_{t,n-1} \in \mathbb{R}^{L \times B \times C}$ is $X^t_{t,n-1}$ with attention score at $t$th timestep in Eq.4. Then, the membrane potential behaviors of a TA-LIF and TA-LIAF layer follow

$$U^{t,n} = H^{t-1,n} + g \left( W^{t,n}, \bar{X}^{t,n-1} \right).$$

The excitation step maps the statistical vector $z$ to a set of temporal-wise input scores. In this regard, the TA module can be deemed as a self-attention function. The main difference is that statistical vectors in the frame-based representation directly correlate with the number of events.

4. Experiments
4.1. Experimental Setup

Datasets. We perform experiments on three kinds of classification datasets, which are all event datasets but are obtained in different ways. The first is DVS128 Gesture[1], which is a gesture recognition dataset capture by DVS cameras. The second is CIFAR10-DVS[19], which is an event-based image classification dataset convert from the static dataset by scanning each sample in front of DVS cameras. The last one is the Spoken Heidelberg Digits (SHD)[6], which is an audio classification dataset convert from audio by software simulation.

Learning. Table 1 lists details for experiments like learning algorithm, loss function, etc. We use the Adam optimizer [16] for accelerating the training process and employ some standard training techniques of deep learning, such as batch normalization, dropout, etc, and the corresponding hyper-parameters and SNN hyper-parameters are shown in Table 2. The network structures of the three tasks are shown in Table 3, and we adopt the same network structure for LIF-SNN and LIAF-SNN in each dataset.

RCS Method. Leave out the time consumption in hardware, event-based system latency $t_{lat}$ only hinges on $dt$ and $T$, i.e., $dt \times T$. Inspired by the random temporal cropping during video recognition method[35], we apply similar data augmentation at the training stage, which is termed as RCS, i.e., select a random $t_0$ (see Fig.1) as the starting point and aggregate consecutive frames. At the test time, we adopt a voting mechanism by following [35], that is, for the given $dt$ and $T$, an event stream is divided into consecutive 10-crops and the length of each one is $t_{lat}$, and the final test result is obtained by accumulating the results of all the individual crops. If the number of frames is less than $10 \times t_{lat}$, we adopt overlap methods in [21], e.g., using 2-crops of 30ms and $t_{lat} = 20\text{ms}$, the crops will cover partially overlapped ranges as [0ms; 20ms] and [10ms; 30ms].

![Figure 2: Temporal-wise attention spiking neural networks. In score vector $d^{n-1}$, different colors represent different attention scores at different timesteps, multiplying them can produce the new input tensor according to Eq.7.](image-url)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CIFAR10-DVS &amp; DVS128 Gesture &amp; SHD Dataset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Representation</td>
<td>Tunable Frames</td>
</tr>
<tr>
<td>Output Latency</td>
<td>$dt \times T$</td>
</tr>
<tr>
<td>Learning Algorithm</td>
<td>STBP[38] &amp; BPTT</td>
</tr>
<tr>
<td>Loss Function</td>
<td>Rate Coding[12]</td>
</tr>
<tr>
<td>Network Structure</td>
<td>CNN-based SNN[12]</td>
</tr>
</tbody>
</table>

Table 2. Hyper-parameter setting.

<table>
<thead>
<tr>
<th>Hyper parameter</th>
<th>DVS128 Gesture</th>
<th>CIFAR10-DVS</th>
<th>SHD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Epoch</td>
<td>100</td>
<td>150</td>
<td>100</td>
</tr>
<tr>
<td>Batch Size</td>
<td>36</td>
<td>64</td>
<td>256</td>
</tr>
<tr>
<td>Learning Rate</td>
<td>$1e^{-4}$</td>
<td>$1e^{-3}$</td>
<td>$1e^{-3}$</td>
</tr>
<tr>
<td>$u_{th}$</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>$c^{-\frac{n}{4}}$</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>$r$</td>
<td>16</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 3. Network structure. MP4-max pooling is $4 \times 4$, nC3-Conv is $3 \times 3$ and has $n$ output feature maps, AP2-average pooling is $2 \times 2$, nFC-Linear layer has $n$ output feature maps.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Network Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>DVS128 Gesture</td>
<td>Input-MP4-64C3-128C3-AP2-128C3-AP2-256FC-11</td>
</tr>
<tr>
<td>CIFAR10-DVS</td>
<td>Input-32C3-AP2-64C3-AP2-128C3-AP2-512C3-AP4-256FC-10</td>
</tr>
<tr>
<td>SHD Dataset</td>
<td>Input-128FC-128FC-20</td>
</tr>
</tbody>
</table>

Figure 3. Ablation study of different TA positions based on DVS128 Gesture in (a) LIF-SNN, and (b) LiAF-SNN. S1, pure SNNs without TA; S2, insert TA only at the input layer; S3, insert TA only at the depth layers (whole network except the input layer); S4, insert TA to the whole network.

4.2. Gesture Recognition

The IBM DVS128 Gesture[1] is an event-based gesture recognition dataset, which has the TR in µs level and $128 \times 128$ spatial resolution. It records 1342 samples of 11 gestures, and each gesture has an average duration of 6 seconds. Note that DVS128 Gesture has two kinds of categories which are 10 and 11 classes, and we select the latter setting that is harder.

Ablation Study of Different TA Positions. The position to insert TA is important, and to evaluate its influence, we design an ablation study of different TA positions, which consist of four: S1, pure SNNs without TA; S2, insert TA only at the input layer; S3, insert TA only at the depth layers (whole network except the input layer); S4, insert TA to the whole network.

Perceptually, smooth interactions in real gesture recognition tasks require systems to respond within 100-200 ms[1]. Based on this requirement, we set $T \in \{30, 60, 90, 120, 150, 180\}$ with $dt = 1\text{ms}$. The impact of the TA positions and simulation timestep $T$ on gesture recognition are shown in Fig.3. For TA positions, we observe that using the S2 (green) or S3 (blue) independently can improve performance in most cases, and S3 achieves the best accuracy. But combining S2 and S3 (i.e., S4, yellow) leads to unstable results, and the accuracy keeps going down when $T$ grows bigger. Besides, when $T$ falls in the first half range, improving $T$ can improve the accuracy slightly, but further enlarging $T$ ($T > 120$) will be helpless.

Experiments of IAP. Inspired by the characteristic of event-triggered computation in LIF-SNNs, we discard the frames with lower attention scores at the inference stage (see Eq.6) for power-saving and term this method as IAP.

It is worth noting that the attention mechanism brings the possibility for the discard operation. To evaluate the effects of input pruning, we set IAP on S2 and S4 since each frame has an attention score in these two cases. To make an intuitive comparison, without attention, we choose a simple input random pruning (IRP) to achieve the same level of power consumption as the baseline. The accuracies of IRP in Fig.4 (a) appear approximative monotone decreasing with the increase of the pruning proportion. However, for Fig.4 (b) and (c), the accuracies do not decrease as the pruning proportion increases at the first half of the pruning proportion. More beneficially, as shown by the dotted circle in Fig.4, when the pruning proportion is 0.5, most of the IAP still maintain high accuracies around 89%, but the IRP accuracies decrease to around 78%. Moreover, as shown in Table 4, the best pruning proportion of the IAP relates to the simulation timestep $T$, and we can get similar or even better performance with almost only half the power and a low-latency (30ms to 180ms) with the TA module compared with using full input.

Ablation Study of RCS and TA. To investigate the influence of RCS and TA, we conduct several ablation studies in Table 5. Because of the stability of the S3 strategy, it will be used for all the rest of experiments in this paper. For the comprehensiveness of the studies, we set multiple-scale $dt \in \{1, 5, 10, 15, 20, 25\}$ with fixed $T = 60$. First, we show the effect of the TA and RCS method individually
Figure 4. Experiments of IAP. We fixed $dt = 1ms$, varying simulation timestep $T \in \{30, 60, 90, 120, 150, 180\}$ and pruning proportion on DVS128 Gesture. From left to right (a) IRP without attention. (b) TA module only used in the input layer with IAP. (c) TA module used in the whole network with IAP. As shown by the dotted circle, when the pruning proportion is 0.5, most of the IAP results still maintain high accuracy around 89%, but the IRP accuracy results decrease to around 78%.

Table 4. Influence of IAP in variant T on parameters, accuracy, and FLOPs. “Param.” means the ratio of increased parameters of TA, “Best Pro.” and “Acc.” reflect the best pruning proportion to keep the accuracy. “FLOPs” means the floating point operations.

<table>
<thead>
<tr>
<th>IAP</th>
<th>T</th>
<th>Param. (%)↑</th>
<th>Best Pro.</th>
<th>Acc. (%)</th>
<th>FLOPs (%)↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2</td>
<td>30</td>
<td>+0.004</td>
<td>0.17</td>
<td>81.95(-1.73)</td>
<td>-16.67</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>+0.018</td>
<td>0.40</td>
<td>86.11(-1.04)</td>
<td>-40.00</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>+0.043</td>
<td>0.30</td>
<td>89.24(-1.39)</td>
<td>-30.00</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>+0.078</td>
<td>0.35</td>
<td>90.99(+0.02)</td>
<td>-35.00</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>+0.123</td>
<td>0.40</td>
<td>89.89(-0.04)</td>
<td>-40.00</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>+0.179</td>
<td>0.40</td>
<td>91.28(+0.31)</td>
<td>-40.00</td>
</tr>
<tr>
<td>S4</td>
<td>30</td>
<td>+0.020</td>
<td>0.34</td>
<td>83.33(-3.48)</td>
<td>-33.33</td>
</tr>
<tr>
<td></td>
<td>60</td>
<td>+0.091</td>
<td>0.24</td>
<td>88.20(-0.34)</td>
<td>-23.33</td>
</tr>
<tr>
<td></td>
<td>90</td>
<td>+0.214</td>
<td>0.32</td>
<td>89.24(-0.69)</td>
<td>-31.11</td>
</tr>
<tr>
<td></td>
<td>120</td>
<td>+0.389</td>
<td>0.50</td>
<td>90.58(-0.05)</td>
<td>-50.00</td>
</tr>
<tr>
<td></td>
<td>150</td>
<td>+0.615</td>
<td>0.35</td>
<td>89.24(+0.35)</td>
<td>-34.67</td>
</tr>
<tr>
<td></td>
<td>180</td>
<td>+0.893</td>
<td>0.35</td>
<td>88.89(+0.00)</td>
<td>-35.00</td>
</tr>
</tbody>
</table>

Table 5. Ablation study of RCS and TA-SNNs. We use the S3 strategy to test multiple-scale TRs with $T = 60$.

<table>
<thead>
<tr>
<th>Model</th>
<th>$dt$</th>
<th>SNN(%) [12]</th>
<th>TA-SNN (%)</th>
<th>SNR (RCS)(%)</th>
<th>TA-SNN (RCS)(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LIF</td>
<td>1ms</td>
<td>71.53</td>
<td>73.25</td>
<td>87.15</td>
<td>90.28(+18.75)</td>
</tr>
<tr>
<td></td>
<td>5ms</td>
<td>87.15</td>
<td>89.24</td>
<td>90.63</td>
<td>93.40</td>
</tr>
<tr>
<td></td>
<td>10ms</td>
<td>91.67</td>
<td>93.40</td>
<td>93.40</td>
<td>94.79</td>
</tr>
<tr>
<td></td>
<td>15ms</td>
<td>93.05</td>
<td>95.49</td>
<td>92.36</td>
<td>95.49</td>
</tr>
<tr>
<td></td>
<td>20ms</td>
<td>92.71</td>
<td>94.44</td>
<td>91.32</td>
<td>94.79</td>
</tr>
<tr>
<td></td>
<td>25ms</td>
<td>93.40</td>
<td>95.14</td>
<td>91.67</td>
<td>95.48</td>
</tr>
<tr>
<td>LIAF</td>
<td>1ms</td>
<td>72.59</td>
<td>74.31</td>
<td>90.97</td>
<td>91.32(+18.73)</td>
</tr>
<tr>
<td></td>
<td>5ms</td>
<td>88.20</td>
<td>89.93</td>
<td>93.06</td>
<td>94.10</td>
</tr>
<tr>
<td></td>
<td>10ms</td>
<td>93.75</td>
<td>95.14</td>
<td>93.75</td>
<td>94.79</td>
</tr>
<tr>
<td></td>
<td>15ms</td>
<td>95.14</td>
<td>96.88</td>
<td>94.10</td>
<td>94.79</td>
</tr>
<tr>
<td></td>
<td>20ms</td>
<td>95.84</td>
<td>97.57</td>
<td>94.10</td>
<td>95.14</td>
</tr>
<tr>
<td></td>
<td>25ms</td>
<td>96.18</td>
<td>98.61</td>
<td>94.44</td>
<td>94.79</td>
</tr>
</tbody>
</table>

Table 6. Ablation experiments on $dt = 10ms$ and $T = 10$ in the CIFAR10-DVS dataset with the S3 strategy.

<table>
<thead>
<tr>
<th></th>
<th>LIF</th>
<th>LIAF</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNN</td>
<td>SNN (RCS)</td>
<td>TA-SNN (RCS)</td>
</tr>
<tr>
<td>54.70%</td>
<td>66.60%</td>
<td>71.10%</td>
</tr>
<tr>
<td>SNR</td>
<td>SNR (RCS)</td>
<td>TA-SNN (RCS)</td>
</tr>
<tr>
<td>69.40%</td>
<td>70.97%</td>
<td>72.00%</td>
</tr>
</tbody>
</table>

4.3. Event-based Image Classification

CIFAR10-DVS[19] is an event-based dataset converted from CIFAR10 by scanning each image with repeated closed-loop movement in front of a DVS. CIFAR10-DVS includes 1000 samples for each category in CIFAR10, and there are in total 10,000 samples, with each one having a duration of 300ms. The temporal and spatial resolutions are $\mu$s and $128 \times 128$, respectively. Unlike gesture recognition, the temporal feature in CIFAR10-DVS may not be dominant[8]. Fig.5 gives examples in CIFAR10-DVS, which can be observed that the temporal correlation between different frames is not obvious. Based on the above analysis, we select moderate parameters that are $T = 10$ and $dt = 10ms$. As shown in Table 6, in these experiments, both RCS and TA-SNNs can improve accuracy.
Figure 5. Examples of consecutive frames in CIFAR10-DVS with $dt = 10\text{ms}$. The movement of images in CIFAR10 is designed by fixed trajectory, and the distance of spatial movement is restricted. Thus, frames at different timestep are similar, and the temporal feature is not the dominant information[8].

Table 7. Experiments on TA-SNNs in SHD dataset with the S3 strategy. For comparison, we fixed $t_{lat}$ here and adopt the same network structure with [43]. Since shorter samples will pad with zeros, the RCS method cannot be used here.

Table 8. Accuracy of models for the DVS128 Gesture, CIFAR10-DVS and SHD Dataset.

4.4. Audio Classification

The SHD dataset[6] is a large spike-based audio classification task that contains 10420 audio samples of spoken digits ranging from zero to nine in English and German languages. A biologically inspired model[6] is used to convert the audio signal into a spike stream, and the data duration ranges from 0.24s to 1.17s. Unlike the four-dimensional event stream generated by the DVS camera, the audio spike stream has only two dimensions, i.e., time and position. As shown in Fig.6, the resolution of time dimension is $\mu$s level, and the position ranges from 0 to 699. We adopt the same data preprocess method in [43], i.e., all samples are fit within a $1\text{s}$ window, where shorter samples are padded with zeros, and longer samples are cut. We use $t_{lat} \in \{200, 600, 900, 1000\}$ with different $dt$ based on S3 strategy. Results in Table 7 demonstrate that TA-SNNs always work in all kinds of parameter combinations, and the bigger $t_{lat}$ is, the higher accuracy we could have. To verify the effectiveness of the TA module, we also insert it into an extra analog-based SNN, i.e., ReLU SRNN[43], and report 90.02% accuracy, which is higher than 88.93% in [43].

4.5. Comparison with Prior Works

We compare our best results of the proposed TA-SNN against various of prior works for event-based data, such as CNN method[1, 32], spike-based SNNs[17, 21, 45, 39, 6, 12, 44], and analog-based SNNs[40, 15, 43], etc. As shown in Table 8, our TA-SNN models achieve the SOTA in various datasets, and the performance of spike-based SNNs and analog-based SNNs have been improved by inserting the TA module. Moreover, comparing with the original SNNs, the number of parameters in TA-SNNs almost has no increase. From the above comparisons, it can be seen that the TA module can help SNN to achieve higher performance with less cost in various tasks, thereby, the TA module will contribute a lot to promote SNNs to practical applications.
5. Discussion

**TA position.** Different TA positions in SNNs have different effects on performance, and the TA module inserted on depth layers (i.e., S3) works better (see Fig.3). This phenomenon is similar to the SE block used in the channel domain, where the SE in deeper layers is slightly better than that in lower layers[13]. Compared with pure SNNs, TA-SNNs based on the S3 can always enhance the network’s ability to extract spatio-temporal features.

**Adaptability of TA module.** One of the most valuable points in the spike-based SNNs is the event-triggered computation feature[11]. However, keeping the event-triggered characteristic also brings difficulty in training since the spike activity is hard to differentiate. Although the STBP algorithm, which can solve the differentiability issue to some extend, appears to be barely satisfactory in deep SNNs, comfortingly, our TA module can improve the accuracy of spike-based SNNs. Moreover, instead of utilizing full input frames, our TA module also brings interesting and important IAP that can get similar or even better performance with only half of the input frames and low latency (30ms to 180ms). This achievement may magnify the advantage of power-saving and exhibit the potential of network performance improvement in deep spike-based SNNs. On the other hand, the TA module also works in analog-based SNNs, which give up the event-triggered characteristic but keep the dynamic characteristics of a biological neuron, and all SOTA results are obtained in this way, however, more power might be needed in return.

**Influence of RCS method.** Prior works mostly used \( t_0 = 0 \) as the starting point in training, while our RCS method selects a random \( t_0 \) (see Fig.1). For the RCS, there is a basic precondition that the content of event streams should have inherent cycles of repetition. Both gesture action in DVS128 Gesture and repeated movement of the image in CIFAR10-DVS satisfy this precondition. As shown in Table 5, RCS works better under short-term \( dt \). However, using RCS with long-term \( dt \) will reduce accuracy, e.g., the accuracy of analog-based SNNs reduces from 98.61% to 94.79% with \( dt = 25ms \). The possible reason is that choosing a long-term aggregation window will destroy the inherent periodicity. In SHD, selecting a random \( t_0 \) may likely cause all the input data to be 0 for a shorter sample, thus RCS does not work here either.

**TR Analysis.** TR is a crucial hyper-parameter for frame-based representation. Current methods usually adopt a long-term TR to make sure SNR is sufficient in each frame. It is indeed useful since all SOTA results in this work are obtained in the long-term TR. However, long-term TR will dilute the advantages of the asynchronicity and sparsity of event streams and increase the output latency. Short-term TR brings high-rate frames that are friendly to high-speed object tracking and low-latency interaction, but the produced low SNR is an intractable issue for getting satisfactory performance. By using RCS and TA, this issue is greatly relaxed in our work. Firstly, the RCS method significantly strengthens short-term TR’s advantages, i.e., the smaller the TR is, the more optional training data we can organize. Secondly, the TA enhances the ability of SNNs to effectively extract spatio-temporal features (see Table 5). Moreover, IAP with short-term TR in spike-based TA-SNN can keep or improve task accuracy. These experiments imply that event streams with short-term TR have a great potential to solve the real-time scenarios with considerable accuracy. Last but not least, the selection of TR also depends on the inherent trait of different tasks or datasets. In our experiments, DVS128 Gesture naturally has affluent repeatability, so short-term TR can obtain an acceptable result. Meanwhile, CIFAR10-DVS has poor temporal features and SHD sample almost has no repeatability, so short-term TR is meaningless for them.

6. Conclusions

In this paper, we innovatively integrate the temporal attention mechanism into SNNs and propose the TA-SNNs that can deal with the event streams more effectively and efficiently than the pure LIF-SNNs while preserving SNNs’ event-triggered feature. Additionally, attention-score-based input pruning technology is used in the inference process, which surprisingly doesn’t cause a significant accuracy loss but saves a large amount of computation. We also propose the RCS method and investigate the performance of TA-SNNs on various datasets in different TRs. The experiment results are provided using TA-SNNs and RCS, and achieve SOTA results in DVS128 Gesture (98.61%), CIFAR10-DVS (72.00%), and SHD (91.08%), verifying the effectiveness of these methods.

We believe that this method will greatly expand people’s imagination of SNNs, guide more advanced deep learning technology into SNNs research, and open up the way for the applications of SNNs. In addition, in future work, this method will also help SNNs to get better performance on hardware. The sparse event-triggered characteristics of SNNs are kept by TA-SNNs, which will be of great significance to improve the performance on the SNNs accelerators.

**Acknowledgment.** This work was partially supported partially by National Key R&D Program of China (2018AAA0102600, 2018YEF0200200), National Natural Science Foundation of China (61876215, 12002254), and Beijing Academy of Artificial Intelligence (BAAI), and the open project of Zhejiang Laboratory, and a grant from the Institute for Guo Qiang of Tsinghua University, and Peng Cheng Laboratory.
References


